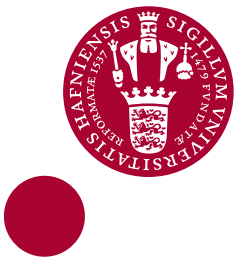

Neural check-worthiness ranking with weak supervision

Finding sentences for fact-checking

Hansen, Casper; Hansen, Christian; Alstrup, Stephen; Simonsen, Jakob Grue; Lioma, Christina

Published in:
The Web Conference 2019 - Companion of the World Wide Web Conference, WWW 2019

DOI:
10.1145/3308560.3316736

Publication date:
2019

Document version
Publisher's PDF, also known as Version of record

Neural Check-Worthiness Ranking with Weak Supervision: Finding Sentences for Fact-Checking

Casper Hansen
Department of Computer Science,
University of Copenhagen

Christian Hansen
Department of Computer Science,
University of Copenhagen

Stephen Alstrup
Department of Computer Science,
University of Copenhagen

Jakob Grue Simonsen
Department of Computer Science,
University of Copenhagen

Christina Lioma
Department of Computer Science,
University of Copenhagen

ABSTRACT

Automatic fact-checking systems detect misinformation, such as fake news, by (i) selecting *check-worthy* sentences for fact-checking, (ii) gathering related information to the sentences, and (iii) inferring the factuality of the sentences. Most prior research on (i) uses hand-crafted features to select check-worthy sentences, and does not explicitly account for the recent finding that the top weighted terms in both check-worthy and non-check-worthy sentences are actually overlapping [15]. Motivated by this, we present a neural check-worthiness sentence ranking model that represents each word in a sentence by *both* its embedding (aiming to capture its semantics) and its syntactic dependencies (aiming to capture its role in modifying the semantics of other terms in the sentence). Our model is an end-to-end trainable neural network for check-worthiness ranking, which is trained on large amounts of unlabelled data through weak supervision. Thorough experimental evaluation against state of the art baselines, with and without weak supervision, shows our model to be superior at all times (+13% in MAP and +28% at various Precision cut-offs from the best baseline with statistical significance). Empirical analysis of the use of weak supervision, word embedding pretraining on domain-specific data, and the use of syntactic dependencies of our model reveals that check-worthy sentences contain notably more identical syntactic dependencies than non-check-worthy sentences.

KEYWORDS

Fact checking; Check worthiness; Deep learning; Weak supervision.

ACM Reference Format:
Casper Hansen, Christian Hansen, Stephen Alstrup, Jakob Grue Simonsen, and Christina Lioma. 2019. Neural Check-Worthiness Ranking with Weak Supervision: Finding Sentences for Fact-Checking. In *Companion Proceedings of the 2019 World Wide Web Conference (WWW'19 Companion), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3308560.3316736

WWW '19 Companion, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6675-5/19/05.
https://doi.org/10.1145/3308560.3316736

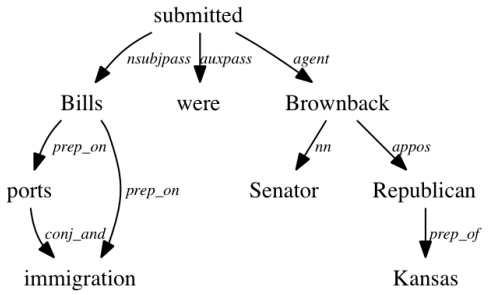

Figure 1: Syntactic dependencies example (from [20]).

1 INTRODUCTION

The fast and wide spread of misinformation (as opposed to true information) on social media [22, 25], and the increasing use of social media as a source of news[1] has turned "fake news" into an important societal problem on a scale that requires automated solutions. An automated fact-checking [21] pipeline typically consists of three steps: (i) selecting *check-worthy* sentences (i.e. sentences that contain check-worthy claims and should be fact-checked), (ii) gathering related information to those sentences, and (iii) using this related information to infer the factuality of each check-worthy sentence. Prior research has so far focused mainly on steps (ii) and (iii), for instance by generating claim-specific queries and querying search engines for relevant supporting information [12], focusing on specific sources such as Twitter [1], or exploiting knowledge graphs from e.g. Wikipedia [5]. These approaches assume an input of check-worthy claims. Considerably less research has focused on detecting such check-worthy claims, that is, determining not whether a sentence is true or not, but whether a sentence contains a check-worthy claim (and should be fact-checked) or not.

Most research on check-worthiness detection uses hand-crafted features, such as bag-of-word representations, sentiment, and embedding averages [7, 8, 10, 19]. In addition, most work in this area does not explicitly account for the recent finding that the top weighted terms in both check-worthy and non-check-worthy sentences are actually overlapping [15], hence compromising the effectiveness of bag-of-word based methods.

Motivated by the above, we present a neural check-worthiness sentence ranking model that uses a dual sentence representation: each word in a sentence is represented by *both* its embedding (aiming to capture the semantics of that word from its context) and its

[1] https://www.reuters.com/article/us-usa-internet-socialmedia/two-thirds-of-american-adults-get-news-from-social-media-survey-idUSKCN1BJ2A8

syntactic dependencies (aiming to capture the role of that word in modifying the semantics of other words in the sentence, see Figure 1). We train the network with these dual representations end-to-end. This allows to learn such descriptive features directly from the input data, rather than relying on predetermined hand-crafted features that may not be useful for the task, and hence to adapt the representations to the domain. To tackle the problem of having very little available training data, we use an existing check-worthiness system to weakly label sentences, and we use this weakly labelled dataset to pretrain our neural network. This is inspired by recent strong results [18, 23] in various information retrieval tasks with few labelled data, but large amounts of unlabelled data.

Thorough experimental evaluation against all state of the art baselines on political speeches from the 2016 U.S. election, shows our model to be superior in all comparisons (+13% in MAP and up to +28% at various Precision cut-offs from the best baseline, with statistical significance). We empirically trace this superior performance to the use of syntactic dependencies in the sentence representation, where we find check-worthy sentences to contain notably more identical syntactic dependencies than non-check-worthy sentences. Further analysis shows that the performance benefits of weak supervision increase with the amount of data used, and that embeddings trained on smaller domain-specific data, as opposed to general purpose embeddings trained on the larger Google News corpus, increase effectiveness. In addition, despite using deep learning (a family of models that is generally considered of low interpretability [24]), the attention weighting used by our model on a word level provides humanly interpretable output, where the parts of the sentence that are important for the check-worthiness prediction can be determined.

We **contribute** a competitive and interpretable end-to-end trainable neural network model for check-worthiness ranking, which uses a dual input representation of both word embeddings and syntactic dependencies. Weak supervision is used to pretrain the network on large amounts of unlabelled data.

2 RELATED WORK

Given a sentence (also referred to as statement) as input, ClaimBuster [8, 9] outputs its check-worthiness score by extracting a set of features (sentiment, statement length, Part-of-Speech (POS) tags, named entities, and tf-idf weighted bag-of-words), and training a SVM classifier on these features to predict check-worthiness. Patwari et al. [19] present a system called TATHYA that is based on similar features, but that also includes as contextual features sentences immediately preceding and succeeding the one being assessed, as well as certain hand-crafted POS patterns. Gencheva et al. [7] also extend the feature set used by ClaimBuster to include more contextual features, such as the sentence's position in the debate text, and whether the debate opponent is mentioned. The work by Gencheva et al. has been extended to both English and Arabic [10]. In the recent CLEF 2018 competition on check-worthiness detection [17], Zou et al. [26] came first by using a large set of features (similarly to the above mentioned work), and doing feature selection with both a χ^2-test and a linear SVM using a L1 regularizer.

Prior work on neural networks for check-worthiness has been done by Konstantinovskiy et al. [14], who use InferSent [6] to derive a universal neural sentence representation and then train a logistic regression classifier on top of that. Similarly to our model, this approach also uses neural sentence embeddings. However, unlike our model, this approach uses *universal* sentence representations, whereas we train our model *end-to-end* to learn the representations directly from the input data, making the learning domain-specific.

In the related domain of sentence factuality detection Jimenez and Li [11] present a neural approach with multiple word embeddings. They artificially generate additional non-factual sentences to be used for training to increase robustness. Similarly to ours, this work also presents neural approaches that combine multiple word representations in order to improve performance. However, whereas the infusion of artificially generated non-factual sentences by Jimenez and Li [11] allows weak supervision of a single class, we obtain weak labels independently of the type of sentence (i.e. not on a single class).

3 NEURAL CHECK-WORTHINESS SENTENCE RANKING

Given a set of sentences as input, the aim is to rank them in descending order of check-worthiness. In order to better differentiate between sentences of varying degree of check-worthiness, We cast this as a ranking problem, as opposed to assigning a binary output to each sentence, following prior work [7, 8, 10] . Note that any ranked output can be made binary using an appropriate threshold, in case a subsequent fact-checking pipeline requires binary labelled sentences.

Given a set of sentences to be ranked, our model learns an end-to-end trained representation of each sentence. We describe first the representation of each word in the sentence (Section 3.1), followed by the neural network architecture (Section 3.2).

3.1 Neural sentence representation

Our model uses a dual sentence representation: each word in a sentence is represented by *both* its embedding and by its syntactic dependencies. The word embedding aims to capture the semantics of that word from its context. Embeddings of this type are generally well understood and have been found effective for check-worthiness detection [14]. The syntactic dependencies of a word aim to capture the role of that word in modifying the semantics of other words in the sentence, for instance by being the subject or predicate of the sentence. We use a syntactic dependency parser [4] to map each word to its dependency (as a tag) in relation to the sentence structure, which we then represent as a one-hot-encoding. Dependency parsing is fast using state of the art tools (approximately 14,000 words per second) [4].

Our motivation is that syntactic dependencies may be important for discriminating between common overlapping top-weighted words in both check-worthy and non-check-worthy sentences. The existence of common overlapping top weighted words in check-worthy and non-check-worthy sentences is reported by Le et al. [15] (see Figure 2 of [15] for examples), and to our knowledge, is not explicitly addressed by any prior check-worthiness approach. We posit that while these common top weighted words may not be distinguishable by their word representation, they may be distinguishable by their syntactic role in the sentence.

Table 1: Statistics of the embedding training, evaluation, and weakly labelled datasets.The evaluation dataset uses binary labels, and the weakly labelled dataset continuous labels in the interval $[0, 1]$.

Dataset	#docs	#sents.	sents. length	unique words	mean label
Embed. tr.	15,059	609,322	16.66	86,244	-
Evaluation	7	2,602	14.04	3,694	0.05
Weakly lab.	161	37,732	16.53	13,314	0.24

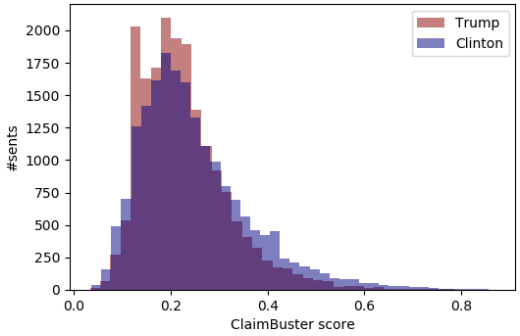

Figure 2: Histogram of the ClaimBuster scores used as the weak labels for each presidential candidate.

3.2 Network architecture

Based on the above, each word in a sentence has two distinct encodings, that together represent the word. We use this double representation of each word as input to a recurrent neural network (RNN) with GRU [3] memory units. The output from each word in the RNN is aggregated using an attention mechanism computed as $\alpha_t = \frac{\exp(\text{score}(h_t))}{\sum_i \exp(\text{score}(h_i))}$, where h_t is the output of the GRU memory cell at time t, and $score(\cdot)$ is a learned function that maps the output to a scalar. The attention-weighted sum is fed to a fully connected layer, from which the output is predicted using a sigmoid activation function. We train the network using the RMSprop optimizer with binary cross entropy as the loss function (details in Section 4.3).

4 EXPERIMENTAL SETUP

We conduct two experiments: (I) we compare our model against state of the art baselines without weak supervision; (II) we use the ClaimBuster API (one of the baselines in experiment I) to weakly label a dataset of unlabelled political speeches (as described in Section 4.2) and we use this weakly labelled data to pretrain the baselines and our model. ClaimBuster API is trained on a non-publicly available dataset and should therefore be considered as a black box baseline.

4.1 Baselines

We compare our model against these baselines (introduced in Section 2), which have yielded state of the art performance at their date of publication: (1) ClaimBuster and its pretrained ClaimBuster API [8], (2) TATHYA [19], and the approaches by (3) Zou et al. [26], (4) Gencheva et al. [7], and (5) Konstantinovskiy et al. [14].

4.2 Data

We use three datasets for three different purposes: (1) the *embedding training* dataset, used to train domain-specific embeddings for our model[2]; (2) the *evaluation* dataset, used to compare our model to the baselines without weak supervision; and (3) the *weakly labelled* dataset, used to compare our model to the baselines with weak supervision. We describe these next (see Table 1 for statistics).

The **Embedding Training Dataset** contains documents related to *all* U.S. elections available through the American Presidency Project[3], e.g. press releases, statements, speeches, and public fundraisers, resulting in 15,059 documents. We use this dataset to pretrain a domain specific word embedding for our model (see Section 5.2).

The **Evaluation Dataset** consists of a total of 2,602 sentences from 7 check-worthiness annotated political speeches from the 2016 U.S election. Out of these 7 speeches, 4 are by Donald Trump and are made available by the CLEF 2018 lab on automatic identification and verification of political claims [17]. The remaining are the inauguration and acceptance speech of Donald Trump and the acceptance speech of Hilary Clinton, and are made available by the authors of ClaimRank [10]. We choose the available PolitiFact annotated labels for this dataset.

The **Weakly Labelled Dataset** consists of all publicly available speeches by Hillary Clinton and Donald Trump from the 2016 U.S. election, which are available through the American Presidency Project. This amounts to 37,732 sentences from 161 speeches not occurring in the evaluation dataset. We use the public API[4] of ClaimBuster [8] to weakly label each sentence in all speeches. The ClaimBuster scores range from 0 to 1 (the higher the score, the more check-worthy the sentence), and are thus continuous as opposed to the binary labels from the evaluation dataset. The distribution of ClaimBuster scores can be seen in Figure 2, where we see that sentences by Hillary Clinton are overall labelled as slightly more check-worthy than those by Donald Trump.

4.3 Tuning

We measure the effectiveness of the ranking outputted by our model and the baselines using mean average precision (MAP), and average precision at multiple cut-offs: P@5, P@10, P@20, and P@R, where R is the number of check-worthy sentences in a given test set. We optimize the MAP when tuning parameters.

We tune and evaluate the approaches using 7-fold cross validation, where the sentences from one speech (see Section 4.2) act as testing data once, while sentences from the remaining speeches are used for training and validation. We use the sentences of each speech as input to the models. In all folds, we set aside 10% of the training data for validation. Each fold-wise evaluation is repeated 5 times with randomly chosen validation data. We report the average score of each metric across the folds and repetitions.

For ClaimBuster [9] and the model by Gencheva et al. [7], we use the best performing setup described in [7]: a double layered fully connected neural network with layer sizes of 200 and 50 respectively. We validate these sizes by keeping the same ratio (4:1) between the layers and test the largest sizes of $\{50, 100, 200, 400\}$,

[2]None of the other approaches support training embeddings.
[3]https://web.archive.org/web/20170606011755/http://www.presidency.ucsb.edu/
[4]https://idir-server2.uta.edu/claimbuster/

test batch sizes of {64, 128, 256, 512}, and use a learning rate of 0.0001. For the approach by Zou et al. [26] we use their multi-layer perceptron model with two hidden layers with sizes of 100 and 8 as per [26]. We validate these sizes by keeping the same ratio (12.5:1) between layers and test the largest sizes of {50, 100, 200, 400} as done earlier. For TATHYA [19] we use the same multi-classifier approach and the same parameters as described in the original paper. For Konstantinovskiy et al. [14] we use the same logistic regression approach as described in the original paper.

For our model, we evaluate the same layer sizes as above with a ratio of 4:1 between the number of neurons in the GRU cell and the single fully connected layer. We train the word embeddings using the word2vec skip-gram model [16] on the embedding training dataset of 15,059 domain specific documents described in Section 4.2. We use standard parameters with a window size of 5 and sample 25 negative samples per word. For the syntactic dependencies of each word, we use the spaCy syntactic dependency parser [4][5].

For the experiment with weak supervision, the ClaimBuster API returns a score from 0 to 1, indicating the degree of check-worthiness estimated by the system. In each fold in the 7-fold cross validation we find the threshold τ that splits the data and makes the fraction of check-worthy samples equal across the training without and with weakly labelled training data. Using these splits we evaluate two thresholding approaches: (1) Binarizing the labels based on τ, and (2) truncating all values larger than τ to the value of τ, and scaling the range $[0, \tau]$ into $[0, 1]$. In the cross validation, our approach performs best with step (2) and the baselines perform best with step (1) (these are the settings we report in Section 5). Note that step (2) can be considered as soft thresholding, as the labels are not binary. The weakly labelled data is used for pretraining the neural models or added to the training data for traditional supervised models.

5 RESULTS

Table 2 shows the results of the experimental comparison of our model to the baselines without and with weak supervision. We see that our model outperforms all baselines across all metrics (with improvements ranging from +9-28%), except P@5 (only without weak supervision) where our model is the second best performing approach. Note that P@k is known to be unstable, especially at small values of k [2, 13]. The best performing baseline is the approach of Konstantinovskiy et al. [14], the only other approach using neural embeddings. This points out the effectiveness of neural word embeddings for this task.

The difference in performance between ClaimBuster and the ClaimBuster API is due solely to the quality of the training data (it is otherwise the same approach) and illustrates the effect of training data quality upon model performance. The fact that our model still notably outperforms the ClaimBuster API baseline shows the benefit of an end-to-end learned representation as opposed to a feature engineered one, for this task.

Only ClaimBuster, the approach of Zou et al. [26], and our model obtain notable improvements from the weakly labelled data (ClaimBuster yields a performance similar to that of the ClaimBuster API). TATHYA [19], and the approaches by Gencheva et al. [7] and Konstantinovskiy et al. [14] do not benefit from weak supervision, most likely because feature-engineering is used as opposed to learning the representation.

5.1 Syntactic dependency similarity between check-worthy sentences

Our syntactic dependencies representations aim to discriminate between top weighted words that are common in check-worthy and non-check-worthy sentences based on the syntactic roles of these words (see Section 3.1). To verify this we compute the average overlap of unique syntactic dependency tags between the following three types of randomly sampled sentence pairs: 1) Pairs of n sampled check-worthy sentences, 2) pairs of n sampled non-check-worthy sentences, and 3) mixed pairs of n sampled check-worthy and n sampled non-check-worthy-sentences. We set $n = 10$ and repeat the computations 1000 times. Table 3 displays the resulting average overlaps and their standard deviation. We observe that check-worthy sentence pairs have the highest average overlap of syntactic dependencies, and non-check-worthy the lowest, but both have a similar standard deviation. This indicates that syntactic dependencies are more similar in check-worthy sentences than in non-check-worthy sentences, and as such constitute a good discriminator between check-worthy and non-check-worthy sentences that otherwise contain an overlap of common top-weighted terms. Note that the average overlap of 7 common syntactic dependencies in check-worthy sentences practically applies to approximately half of the words in those sentences (the average sentence length is 14.04 in that dataset – see Table 1). Mixed sentences (both check-worthy and non-check worthy) have an average overlap in between that of check-worthy and non-check-worthy sentences, but with a larger standard deviation, indicating that syntactic dependencies from this mixed group act as a mixed and less stable discriminating signal.

As an example of the overlap problem of common top-weighted terms, consider the check-worthy sentence "*Since president Obama came into office another two million hispanic americans have fallen into poverty*" and non-check-worthy sentence "*I'm running to be a president for all americans*". In these cases the words *president* and *americans* are both important to describe the content, but have different syntactic dependencies (`compound`/`attr` and `nsubj`/`pobj`, respectively).

5.2 Impact of pretrained word embeddings

Our model is the only approach in Table 2 to have word embeddings trained on a domain specific dataset. All other approaches either use no word embeddings (ClaimBuster [8] and TATHYA [19]), use global word embedding averages (Zou et al. [26] and Gencheva et al. [7]), or use a universally trained sentence representation based on global embeddings (Konstantinovskiy et al. [14]). To isolate the effect of these domain-specific trained embeddings upon the performance of our model, we run our method as described in Section 4.3 but vary the pretraining of the embeddings as follows: 1) using no embeddings at all; 2) using randomly initialized embeddings which are not pretrained; 3) using general purpose embeddings pretrained with word2vec on Google News[7]; 4) using our pretrained domain

[5]The syntactic dependency parser is available at https://spacy.io/

[7]https://code.google.com/archive/p/word2vec/

Table 2: Effectiveness of sentence check-worthiness ranking without and with weak supervision. ▲ marks statistically significant improvements with respect to the overall best baseline at the 0.05 level using a paired two tailed t-test. △ marks statistically significant improvements with respect to the best overall approach without weak supervision at the 0.05 level using a paired two tailed t-test. Significance comparisons are done on the average metric-wise performance in each of the 5 repeated runs.

	Without Weak Supervision					With Weak Supervision				
	MAP	P@5	P@10	P@20	P@R	MAP	P@5	P@10	P@20	P@R
ClaimBuster API[6]	0.230	0.219	0.176	0.159	0.138	-	-	-	-	-
TATHYA [19]	0.136	0.072	0.062	0.074	0.039	0.147	0.061	0.047	0.060	0.043
ClaimBuster [9]	0.176	0.170	0.112	0.105	0.078	0.224	0.198	0.147	0.138	0.121
Zou et al. [26]	0.187	0.143	0.105	0.099	0.086	0.212	0.171	0.111	0.121	0.097
Gencheva et al. [7]	0.238	0.276	0.170	0.153	0.123	0.236	0.222	0.143	0.138	0.113
Konstantinovskiy et al. [14]	0.267	**0.314**	0.186	0.178	0.149	0.233	0.220	0.146	0.142	0.113
Our model	**0.278**	0.291	**0.194**	**0.193**	**0.159**	**0.302**$^{▲△}$	**0.344**$^{▲}$	**0.238**$^{▲△}$	**0.218**$^{▲△}$	**0.189**$^{▲△}$

Table 3: Average overlap of syntactic dependency tags and its standard deviation between three types of sentence pairs.

Sentence pairs	Average Overlap	Standard deviation
Check-worthy	7.00	1.03
Non-check-worthy	4.74	1.08
Mixed	5.64	2.87

Table 4: Performance per type of embedding pretraining. The last row shows the performance without the syntactic dependency parsing.

Emb. pretrain	MAP	P@5	P@10	P@20	P@R
No embed.	0.184	0.153	0.116	0.103	0.087
No pretraining	0.237	0.230	0.156	0.148	0.130
Google News	0.268	0.262	0.178	0.184	0.143
Politics	**0.302**	**0.344**	**0.238**	**0.218**	**0.189**
Politics w/o syn. dep.	0.285	0.290	0.209	0.202	0.167

specific embeddings as described in Section 4.2. Table 4 shows the results when varying the embedding strategy. We see that domain specific embeddings (Politics) obtains large improvements – compared to the general purpose embedding – with improvements up to +12-34%. The last row of Table 4 shows the performance without the syntactic dependency parsing (i.e., only the word embedding), which highlights the large performance increase from the syntactic dependency parsing. As expected, no pretraining of the embeddings leads to much lower performance, though MAP is still comparable to most baselines, except for Konstantinovskiy et al. [14]. Not using embeddings at all severely drops overall effectiveness. Collectively these findings show that performance benefits more from training embeddings on smaller, yet domain-specific, data than on much larger but general domain data.

5.3 Effect of varying the amount of weakly labelled data

We analyse how our model, when used with weak supervision, scales with the amount of weakly labelled data, by reporting performance across the range of 0% to 100% weakly labelled data with

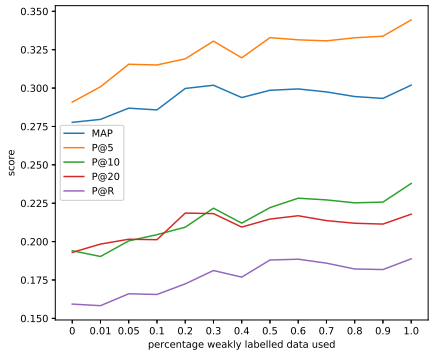

Figure 3: Impact of the amount of weakly labelled data upon our model (0 corresponds to no weakly labelled data).

10% increments. At each step we repeat the experiment 5 times with randomly sampled weakly labelled data, and report the average score. Figure 3 displays performance as a function of the percentage of weakly labelled data. As expected, the scores of all metrics generally increase as the amount of data increases. However, the largest increases happen in the first 50% of the data, and then small increases or stagnation for the remaining range up to around 90%. The performance drop at 40% may be explained by the limited number of repetitions of the sampling process, which was done due to runtime considerations. We expect that a larger number of repetitions would smooth out this slight drop.

5.4 Model interpretability

Check-worthiness detection can be part of semi-automatic or even fully manual fact checking processes, to filter claims that human fact checkers should investigate. In such cases, the output of check-worthiness detection should be easily interpretable by humans. Our model, despite being a deep learning model (generally considered to have low interpretability [24]) – provides easily interpretable output to humans through the attention mechanism that is computed on a word level (see Section 3.2). This score can be used to highlight which parts of a sentence are important for the prediction of check-worthiness. Table 5 illustrates this with a sample of true and false predictions made by our model. We see that sentences with high

predicted scores (both correctly and incorrectly predicted as check-worthy) contain a quantifiable fact consisting of a relative large number, e.g. a large amount of money (*trillion dollars, $800 billion*), a high percentage (*60 percent*), or a large collection of entities (*nearly all other presidents combined*). Sentences with low predicted check-worthiness are more varied, but generally either lack a quantifiable element or are generally vague (*buy American and hire American, fix the system, no patience for injustice*). We can also use the model to find seemingly incorrectly labeled sentences, as e.g. the non-check-worthy labelled sentences with high predictions could indeed be labelled as check-worthy instead, e.g. *"our trade deficit in goods with world last year was nearly $800 billion dollars"*.

Table 5: Check-worthy and non-check-worthy samples with high and low predictions ($\tilde{Y}$) and ground truth labels (Y). Words are colored according to attention weight: the deeper the shade of red, the larger the attention score assigned.

Y	$\tilde{Y}$	Sentence
1	0.96	america has spent approximately six trillion dollars in the middle east , all this while our infrastructure at home is crumbling .
1	0.95	today , our total business tax rate is 60 percent higher than our average foreign competitor in the developed world .
1	0.26	its the same reason why she wo nt take responsibility for her central role in unleashing isis all over the world .
1	0.22	we will follow two simple rules ; buy american and hire american .
0	0.04	millions of democrats will join our movement , because we are going to fix the system so it works fairly and justly for all americans .
0	0.05	i have no patience for injustice .
0	0.94	in the last eight years , the past administration has put on more new debt than nearly all of the other presidents combined .
0	0.95	our trade deficit in goods with the world last year was nearly $ 800 billion dollars .

6 CONCLUSION

We have presented an end-to-end trainable neural network model for ranking check-worthy sentences. The model is pretrained via weak supervision from a large collection of unlabelled data and employs a recurrent neural network with a double representation of each word using domain specific word embeddings and the syntactic dependency parsing of a sentence. We evaluate our model on check-worthy annotated political speeches from the U.S. 2016 presidential election (following the same setting as in the official CLEF 2018 competition on check-worthiness detection [17] but using even more data). Our model does not make use of specialized hand-crafted features as most related work [7, 8, 10, 19], but instead adapts the model and its representation to the domain by being trained in an end-to-end fashion. Thus, our model should by design be able to adapt to other check-worthiness tasks with results depending on the type of discourse and rhetoric. Our model effectively incorporates weak supervision: using an existing check-worthiness ranking system to weakly label political speeches significantly improved performance. Overall, our model outperforms all state of the art baselines in mean average precision and precision at different cut offs, with statistically significant +9-28% gains from the best performing baseline. Further analysis revealed the significance of domain specific word embeddings, compared to traditional general purpose embeddings, and how check-worthy sentences share a syntactic similar structure than non-check-worthy sentences.

Future work consists of investigating further multiple weak signals and incorporating text discourse context into the model.

ACKNOWLEDGMENTS

Partly funded by Innovationsfonden DK, DABAI (5153-00004A), and AMAOS (7076-00121B).

REFERENCES

[1] Mouhamadou Lamine Ba, Laure Berti-Equille, Kushal Shah, and Hossam M Hammady. 2016. Vera: A platform for veracity estimation over web data. In *International Conference Companion on World Wide Web*. 159–162.
[2] Chris Buckley and Ellen M. Voorhees. 2000. Evaluating Evaluation Measure Stability. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 33–40.
[3] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. *Syntax, Semantics and Structure in Statistical Translation* (2014), 103–111.
[4] Jinho D Choi, Joel Tetreault, and Amanda Stent. 2015. It depends: Dependency parser comparison using a web-based evaluation tool. In *Annual Meeting of the Association for Computational Linguistics*. 387–396.
[5] Giovanni Luca Ciampaglia, Prashant Shiralkar, Luis M Rocha, Johan Bollen, Filippo Menczer, and Alessandro Flammini. 2015. Computational fact checking from knowledge networks. *PloS one* 10, 6 (2015).
[6] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Conference on Empirical Methods in Natural Language Processing*. 670–680.
[7] Pepa Gencheva, Preslav Nakov, Lluís Màrquez, Alberto Barrón-Cedeño, and Ivan Koychev. 2017. A Context-Aware Approach for Detecting Worth-Checking Claims in Political Debates. In *International Conference Recent Advances in Natural Language Processing*. 267–276.
[8] Naeemul Hassan, Fatma Arslan, Chengkai Li, and Mark Tremayne. 2017. Toward automated fact-checking: Detecting check-worthy factual claims by ClaimBuster. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1803–1812.
[9] Naeemul Hassan, Gensheng Zhang, Fatma Arslan, Josue Caraballo, et al. 2017. ClaimBuster: the first-ever end-to-end fact-checking system. *Proceedings of the VLDB Endowment* 10, 12 (2017), 1945–1948.
[10] Israa Jaradat, Pepa Gencheva, Alberto Barrón-Cedeño, Lluís Màrquez, and Preslav Nakov. 2018. ClaimRank: Detecting Check-Worthy Claims in Arabic and English. In *Conference of the North American Chapter of the Association for Computational Linguistics*. 26–30.
[11] Damian Jimenez and Chengkai Li. 2018. An Empirical Study on Identifying Sentences with Salient Factual Statements. In *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.
[12] Georgi Karadzhov, Preslav Nakov, Lluís Màrquez, Alberto Barrón-Cedeño, and Ivan Koychev. 2017. Fully Automated Fact Checking Using External Sources. In *International Conference Recent Advances in Natural Language Processing*. 344–353.
[13] Diane Kelly, Xin Fu, and Chirag Shah. 2010. Effects of position and number of relevant documents retrieved on users' evaluations of system performance. *Transactions on Information Systems (TOIS)* 28, 2 (2010), 9.
[14] Lev Konstantinovskiy, Oliver Price, Mevan Babakar, and Arkaitz Zubiaga. 2018. Towards Automated Factchecking: Developing an Annotation Schema and Benchmark for Consistent Automated Claim Detection. http://arxiv.org/abs/1809.08193
[15] Dieu-Thu Le, Ngoc Thang Vu, and André Blessing. 2016. Towards a text analysis system for political debates. In *LaTeCH@ACL*.
[16] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[17] Preslav Nakov, Alberto Barrón-Cedeno, Tamer Elsayed, Reem Suwaileh, et al. 2018. Overview of the CLEF-2018 CheckThat! Lab on automatic identification and verification of political claims. In *International Conference of the CLEF Association*.

[18] Yifan Nie, Alessandro Sordoni, and Jian-Yun Nie. 2018. Multi-level abstraction convolutional model with weak supervision for information retrieval. In *International ACM SIGIR Conference on Research & Development in Information Retrieval*. 985–988.

[19] Ayush Patwari, Dan Goldwasser, and Saurabh Bagchi. 2017. TATHYA: A multi-classifier system for detecting check-worthy statements in political debates. In *ACM on Conference on Information and Knowledge Management*. 2259–2262.

[20] Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, et al. 2014. A Gold Standard Dependency Corpus for English. In *International Conference on Language Resources and Evaluation*. 2897–2904.

[21] James Thorne and Andreas Vlachos. 2018. Automated Fact Checking: Task Formulations, Methods and Future Directions. In *Proceedings of the 27th International Conference on Computational Linguistics*. 3346–3359.

[22] Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science* 359, 6380 (2018), 1146–1151.

[23] Hamed Zamani, W. Bruce Croft, and J. Shane Culpepper. 2018. Neural Query Performance Prediction Using Weak Supervision from Multiple Signals. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 105–114.

[24] Quanshi Zhang and Song-Chun Zhu. 2018. Visual interpretability for deep learning: a survey. *Frontiers of IT & EE* 19, 1 (2018), 27–39.

[25] Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018. Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys (CSUR)* 51, 2 (2018), 32.

[26] Chaoyuan Zuo, Ayla Ida Karakas, and Ritwik Banerjee. 2018. A hybrid recognition system for check-worthy claims using heuristics and supervised learning. In *CLEF 2018 Working Notes* (10 ed.). CEUR-WS.org.